System Stability: A Proxy for "Graceful Degradation"

Dr. **James P. Ignizio**, Ph.D., US Army Logistics Management College
e-mail: ignizio@virginia.edu

James P. Ignizio, Ph.D.

Within the military sector the notion of "graceful degradation" is universally accepted. Military systems (e.g., weapons, force mixes, communication links, air defense systems and even a "system" of strategies and tactics) should, it is agreed, *gracefully degrade* (e.g., under hostile conditions, or random failures, or variations in mission, or changes/modifications in personnel and equipment) — rather than collapse like a house of cards.

Unfortunately, there is no agreement as to how one defines *graceful degradation*, or how it is measured. Furthermore, and perhaps most unsettling, the attributes of *optimality* and *graceful degradation* may — if the hypothesis of this article holds — actually be in opposition. To illustrate this phenomenon, consider the simple "block world" problem depicted in Figure 1.

The stack of blocks on the left side of Figure 1 is unequivocally optimal *in the sense of being the taller* of the two stacks. However, while the stack on the right side of the figure is shorter, less impressive in appearance, and "sub-optimal;" it is also clearly far more stable. Given the choice between attempting to stand on either of the two stacks, most people would select the suboptimal stack. Clearly, something more than the height of the stack is important — something difficult to put into words or formulas.

In this article I explore the very real possibility that "optimal" solutions may be invariably unstable — wherein stability is defined as: ***"the measure of both the speed and ease by which a given solution 'de-evolves' (degrades) to some minimally acceptable level."*** *In the case of the "block world" illustration given* earlier, it should be apparent that the "optimal" stack is likely to collapse easier and faster than the shorter stack.

In addition, what would appear (based on results thus far) to be a practical and effective approach for the assessment of the stability of any given solution is presented. Its performance on a number of real world problems is described. I then contrast the inherent stability of solutions as produced by traditional optimization with those developed by evolutionary means (e.g., genetic algorithms, evolutionary computation).

While the results of my investigation have thus far upheld my *hypothesis* (i.e., that *unstable solutions de-evolve faster and easier than stable ones*), it should be made clear that these results have been limited to the (intensive) investigation of but nine problems (albeit real problems and real data). Since there would appear to be no way to investigate the phenomenon of stability other than empirically, it is hoped that this article motivates others to evaluate the process on their own set of (real world) problems.

Conventional wisdom holds that one should always strive for solutions that are optimal, or at least "near optimal." The idea of *intentionally* developing *non-optimal* solutions is, in itself, an anathema to the operations research profession — and particularly to the academic community. However, as a practicing OR analyst for more than 30 years, I have noted that a surprising number of "optimal solutions" to real world problems have led to unexpected and troubling consequences. Specifically, while such solutions may be optimal on paper, they prove to be problematic when actually implemented.

Just two of the many indications of instability of optimal solutions I have personally encountered are listed below:

SAM-D: SAM-D was the original designation for what is now known as the Patriot air defense system. In the late nineteen-sixties I was tasked with the development of a scheme for the deployment of the elements of such a system. In other words, to produce a method to locate the sites for the missile launchers and radars so as to minimize "leakage" (i.e., protect a region of airspace from attack by enemy aircraft). It was discovered that a branch-and-bound approach, which guaranteed optimal or near optimal solutions, also resulted in deployments that were extremely unstable. For example, if some combination of system elements (e.g., positions, weather, target signature) were changed — even slightly, air defense performance would often suffer a dramatic reduction. Yet, when deployed by means of a heuristic method, the results were quite stable - at the cost of but a very slight reduction in the "optimality" of the solution.

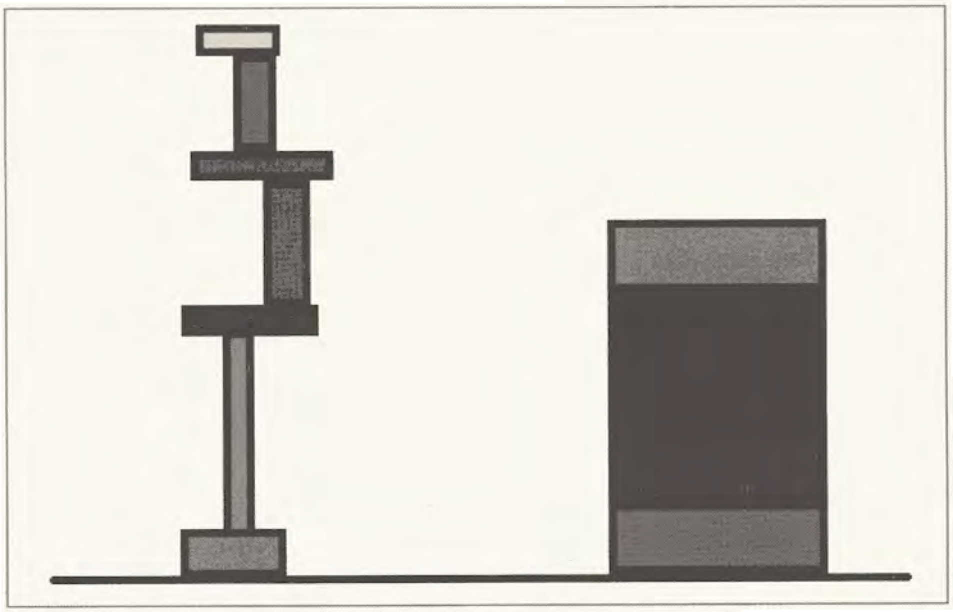

Figure 1: "Optimal" and "Suboptimal" Stacks

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **MAR 1999**	2. REPORT TYPE	3. DATES COVERED **00-00-1999 to 00-00-1999**
4. TITLE AND SUBTITLE **System Stability: A Proxy for 'Graceful Degradation'**		5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Army Logistics Management College, 562 Quarters Rd., Building 12420 ,Fort Lee,VA,23801-1705**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S) 11. SPONSOR/MONITOR'S REPORT NUMBER(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release; distribution unlimited**		
13. SUPPLEMENTARY NOTES		
14. ABSTRACT		
15. SUBJECT TERMS		

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**4**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

Torpedo Acoustic Arrays: Acoustic arrays for torpedoes (or electromagnetic arrays for radars) consist of a number of transducers, acting as an ensemble. These transducers are to be located in such a manner, and delivered power of such amplitude and phase, as to produce a beam pattern of a specified shape. In essence, the array design problem is one of combinatorial optimization, and may be formulated and solved by conventional methods for an optimal solution. However, once these "optimal" acoustic arrays are constructed and tested (typically in a water chamber, under a variety of conditions), the actual performance may be, in a word, *awful*. Just a few seemingly insignificant changes in a combination of parameters (e.g., a slight decrease in temperature coupled with a small increase in pressure) can, and often does, result in sudden and dramatic degradation of the beam pattern. As such, a design that is optimal on paper may well be impractical for application.

While numerous other instances could be cited, all have a similar property. That is, optimal solutions, even when accompanied by intensive (but conventional) sensitivity analysis, are often found to be highly unstable. Yet solutions that are heuristically derived, and clearly sub-optimal, can be as "solid as a rock."

Based upon these experiences, coupled with a long-time interest in heuristic methods (particularly those that mimic evolution, such as genetic algorithms), I sought to test the following hypothesis:

Inherently unstable solutions will de-evolve *faster* and *easier than stable solutions.*

To determine the validity of this hypothesis, a "reverse" genetic algorithm was developed. That is, instead of starting from a poor (or randomly selected) solution and seeking to evolve to more fit solutions, my algorithm begins with any given solution (e.g., the solution to be tested for stability) and *de-evolves* to less fit solutions. It was (as implied in the hypothesis) my conjecture that an unstable solution would de-evolve (e.g., collapse) in fewer generations than a stable one.

For those unfamiliar with genetic algorithms, a list of resources is provided at the end of this article.

In brief, a genetic algorithm proceeds by first coding a trial solution into a "chromosome" (e.g., a pattern of zeros and ones that serve to represent the values of the decision variables). Next, a population of solutions (chromosomes) is generated (typically randomly). From there, a parallel and probabilistic search procedure ensues. Solutions are evaluated for their "fitness." Those that are most fit are given a higher likelihood of being placed into a "mating pool." The mating pool is generated, stochastically, and solutions "exchange portions of their chromosomes" with their mates so as to produce new solutions (e.g., an exchange of a segment of zeros and ones in one chromosome, or coding, with those in another). Mutation (e.g., the "flipping" of a zero to a one, or vice-versa) then takes place (albeit with a very low probability) and the resulting set of solutions represent the "next generation." The process repeats until a given termination criterion is reached.

Since my intent is to find out how easy and fast a given solution *de-evolves* (rather than *evolves*), my genetic algorithm starts with the solution (chromosome) to be tested and works backwards. The pseudo-code for the de-evolution algorithm is provided below.

```
procedure De-Evolve:
begin
t=0
select chromosome, C(t)
perturb C(t) [to generate initial
  population, P(t)]
fitness P(t)
until (done)
        t = t + 1
        select P(t) from P(t-1)
        crossover P(t)
        mutate P(t)
        fitness P(t)
end
```

It is probably easiest to explain the process, and its interpretation, by example. Thus, first consider the results shown in Figure 2. In this figure, two solutions to the design of the "backbone" network for a telecommunications system are depicted. The diagnol symbol, to the far left of the figure, depicts the optimal solution to the problem (producing a normalized value of 100 for the measure of messages per unit time). The box symbol on the far left represents a solution derived via a genetic algorithm. Its value is 94 units, some 6 percent less than that of the optimal solution. Let us assume that the *minimally acceptable level* of system performance is 60 units. Using the de-evolution algorithm, we then determine just how many generations are required for both solutions to degrade to the level of 60 message units.

Examining Figure 2, we see that the "sub-optimal" solution takes roughly *twice* as long (more than 20 generations) to degrade to the minimally acceptable level than does the optimal solution (which de-evolves to the minimally acceptable level in just 10 generations). In other words, the sub-optimal solution is apparently more stable than the optimal solution.

Of course, the result shown in Figure 2 might just be a fluke. After all, we are dealing with a stochastic search process. Consequently, the de-evolution algorithm is repeated numerous times (using different random number seeds) and the results presented in a histogram like that shown in Figure 3.

In Figure 3 it is apparent that the optimal solution (i.e., the de-evolutions shown to the left of the vertical dashed line) does in general de-evolve faster and easier than the suboptimal solution (those to the right of the dashed line, as originally derived by means of a genetic algorithm). Results of many more de-evolutions, as well as the investigation of some eight other real world problems confirmed this observation. One of these eight problems was that of the siting of the elements of the Patriot Air Defense System.

Using data from the original SAM-D air defense study, I developed a number of different siting schemes (e.g., location coordinates for the radars and missile launchers) for the air defense system. One of the solutions was optimal, having been derived via a tedious and time consuming implicit enumeration method. Another solution was derived by means of a genetic algorithm. All other solutions (eight more in total) were developed by various perturbations of these two results. The problem parameters were that of the precise coordinates where each element of the air defense systems was located, the estimates of terrain topology and weather parameters. The fitness of the solution was based upon the amount of airspace covered by the air defense system. The results are shown in Table 1.

(See **STABILITY,** *p. 8)*

STABILITY
(continued from p. 7)

Examining Table 1, we note that the global optimal solution is air defense system A; the one developed via implicit enumeration. Solution F is the solution developed by means of a genetic algorithm. Not only is solution F not optimal, it is actually dominated by solution C. That is, C is both cheaper and has a higher measure of effectiveness than F. If we were to stop our analysis at this point, solution F wouldn't look very attractive at all.

However, after applying the de-evolution algorithm to all ten solutions (and after repeating the process numerous times, using different random number schemes), it was found that solution F was, far and away, the most stable solution of all those tested. Solution A, the "optimal" scheme, fell apart with only slight changes in various combinations of model parameters.

Thus we are left with the following choice. We may either pick the global "optimal" solution (or any solution with a higher effectiveness to cost ratio than solution F), and suffer the consequences of moderate to extreme instability, or select the rock-steady solution F — at a slight reduction in the efficiency to cost ratio. When one considers the fact that the minor difference in effectiveness to cost could well be a result of errors in data (and all real world problems have such errors), solution F starts looking *very* attractive.

As mentioned earlier, results on nine different, real world problems, have all substantiated my original hypothesis. Does this mean that optimal solutions are always unstable? Or that solutions derived by genetic algorithms are always more stable than optimal solutions?

The short answer is no. Just nine experiments obviously cannot prove or disprove the hypothesis. However, these nine, highly consistent results *should* make the OR community take pause. Hopefully this article will encourage others to investigate this matter also. The more empirical evidence in support of this article's hypothesis, the more wary any OR practitioner should be of unquestionably accepting the doctrine of "optimality."

(See **STABILITY,** *p. 27)*

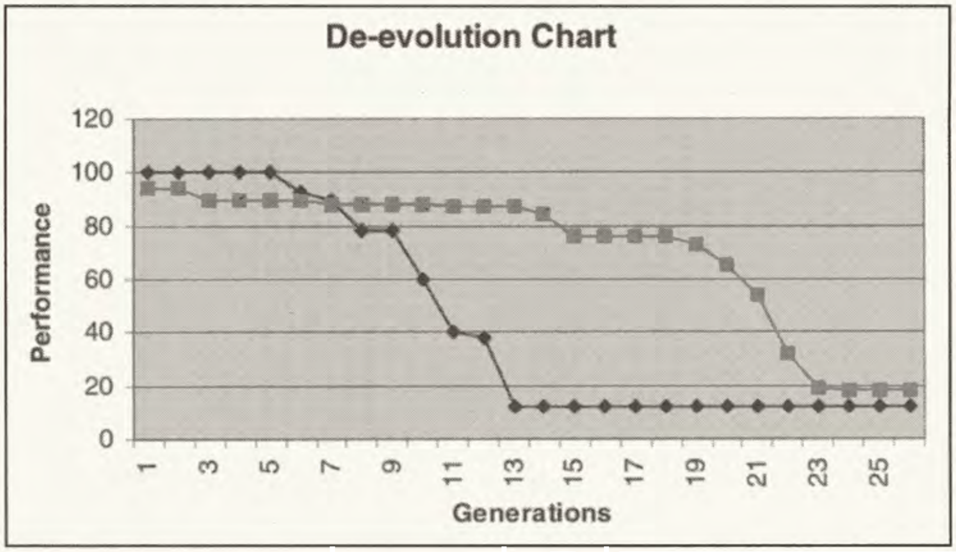

Figure 2: The de-evolution of two different solutions to a network design problem

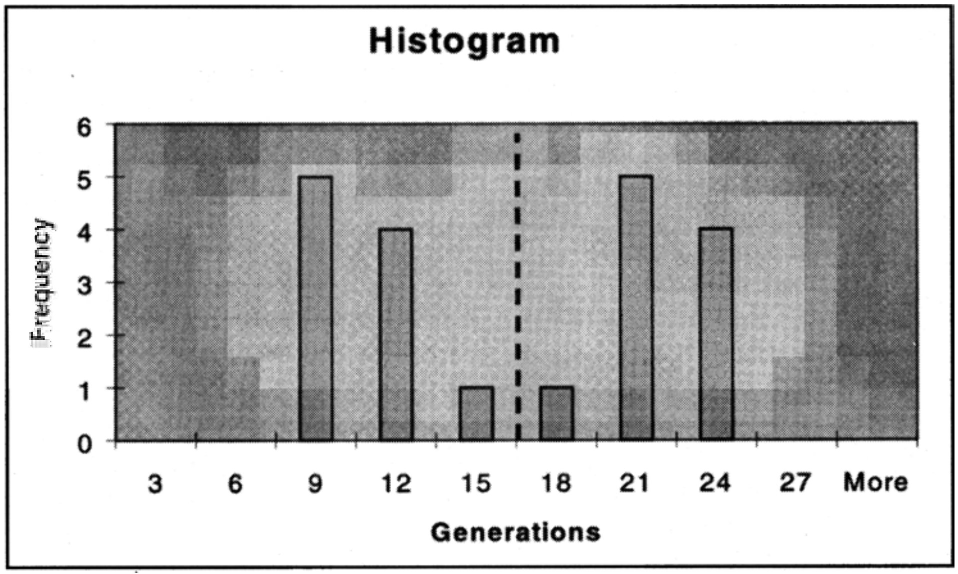

Figure 3: Histogram of De-evolutions

Candidate System	Cost	Effectiveness	Effectiveness to cost ratio
A	50	80	1.6
B	48	76.32	1.59
C	52	84.42	1.585
D	54	84.78	1.57
E	49	76.44	1.56
F	53	81.62	1.54
G	56	84	1.5
H	57	79.8	1.4
I	56	67.2	1.2
J	59	64.9	1.1

Table 1: Candidate Air Defense Systems

sioned as an Ensign in 1970, following graduation from the College of New Rochelle with a Bachelor of Arts degree in Mathematics. She also holds a Masters degree, with distinction, in Operations Research from the Naval Postgraduate School.

Her initial assignment was to the Naval Space Surveillance Systems in Dahlgren, Virginia, where she qualified as a Command Center Officer and orbital analyst. Following a tour on the staff of the Commander in Chief of the Pacific Fleet, she served at the Bureau of Naval Personnel as the Placement Officer for graduate education and service college students.

From 1980 to 1982, Vice Admiral Tracey served as an extended planning analyst in the Systems Analysis division on the Chief of Naval Operations staff. She served as Executive Officer of the Naval Recruiting District in Buffalo, New York, until 1984, when she was assigned as a manpower and personnel analyst in the Program Appraisal division of the Chief of Naval Operations staff.

Vice Admiral Tracey commanded the Naval Technical Training Center at Treasure Island from 1986 to 1988. She then headed the Enlisted Plans and Community Management Branch on the Chief of Naval Personnel's staff for two years. She assumed command of Naval Station Long Beach, California, in 1990.

Upon completion of her command tour, Vice Admiral Tracey reported as a Fellow with the Chief of Naval Operations Strategic Studies Group at the Naval War College. Vice Admiral Tracey was assigned as the Director for Manpower and Personnel, J-1, on the Joint Staff from July 1993 to June 1995. From June 1995 to July 1996 she served as Commander, Naval Training Center, Great Lakes. She assumed the duties of Chief of Naval Education and Training and Director of Naval Training for the Chief of Naval Operations, 10 July 1996.

The Admiral's personal decorations include the Defense Distinguished Service Medal, three Legion of Merit awards and three Meritorious Service Medals.

Vice Admiral Tracey's husband, Richard Metzer, is a former naval flight officer from Pengilly, Minnesota. ✪

STABILITY

(continued from p. 8)

Another question that arises is: If solutions obtained by means of genetic algorithms are indeed more stable than those derived by conventional, optimal seeking methods, why is this so? It is my guess that the inherent stability of the solutions derived by algorithms that emulate evolution is due to the implicit role that stability plays in evolution. As populations (e.g., of plants and animals) evolve, they invariably tend toward stability as well as fitness. Mother Nature simply does not tolerate highly unstable populations of any species. By mimicking the evolutionary process, it would seem that genetic algorithms also provide solutions that are inherently stable.

Acknowledgement

The author would like to express his gratitude and appreciation to the U.S. Army Logistics Management College. ALMC provided the support necessary to conduct the experiments described within the article.

Resources

1 Davis, Lawrence (editor). (1991) *Handbook of Genetic Algorithms*, New York: Van Nostrand Reinhold.

2 Zalzala, A.M.S. and Fleming, P.J. (editors). (1997) *Genetic Algorithms in Engineering Systems*," London: The Institution of Electrical Engineers.

3 Ignizio, J. P. and Cavalier, T. M. *Linear Programming*, Englewood Cliffs, NJ, Prentice-Hall Publishing, 1994.

4 Ignizio, J.P. "Integrating, Cost, Effectiveness, and Stability," accepted in *Acquisition Review Quarterly*.

5 Ignizio, J.P., Wiemann, K.W., and Hughes, W.J. (1987) "Sonar Array Element Location: A Hybrid Expert Systems Approach," *European Journal of Operational Research*, Vol. 32, pp. 76-85.

6 Perlis, J.H. and Ignizio, J.P. (1980) "Stability Analysis: An Approach to the Evaluation of System Design," *Cybernetics and Systems*, Vol. 11, pp. 87-103.

Biography

Education

Ph.D., Operations Research and Industrial Engineering, Virginia Polytechnic Institute and State University

Experience

Professor, Department of Systems Engineering, University of Virginia

Chairman, Department of Systems Engineering, University of Virginia

Professor and Chairman, Department of Systems Engineering, University of Houston

Professor, Department of Industrial Engineering, Pennsylvania State University

Research Professor, University of Alabama, Huntsville

Authored

Eight Books, including ***Linear Programming in Single and Multiple Objectives*** *and* ***An Introduction to Expert Systems***

250+ technical publications

Accomplishments

1963 *Development of Nonlinear Goal Programming*

1968-1972 *Development of the Multidimensional Dual, Integer Goal Programming, and Sequential Multiplex*

1980-1990 *Rulebase Design in Knowledge-Based Systems*

Recent *Hybrid Systems in AI and OR, Ontogenic Neural Networks, Stability Analysis in Large-Scale Systems*

First Engineering Design Application of Goal Programming

Recipient of the first Hartford Prize ✪

MAS PRESIDENT

(continued from p. 4)

work. This material needs to span the theoretical, practical, managerial and organizational. We have already begun this. Our 1MAS meeting produced proceedings that we are currently distributing to the attendees and will shortly be economically available for purchase by everyone. Dr. **J. P. Ballenger**, our Vice President, has begun to collect a volume of essays from our seniors on the future needs of our profession, tentatively entitled *War After Next*. Dr. **Jim Taylor**, NPS, has developed some intriguing panel discussions for the Cincinnati meeting, and we hope to have a MAS Plenary there. Also, we are planning some educational sessions at 2MAS.

In future columns: First MAS Survey and details of 2MAS. ✪